PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2005-135024
(43)Date of publication of application : 26.05.2005

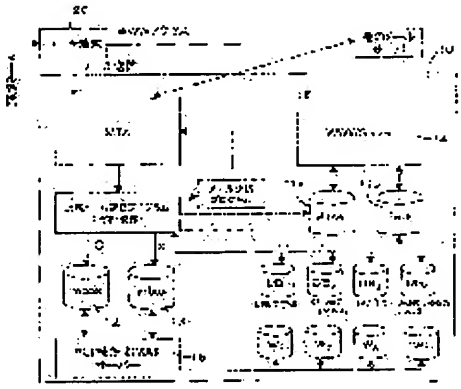
(51)Int.Cl. G06F 13/00

(21)Application number : 2003-367895
(22)Date of filing : 28.10.2003

(71)Applicant : ANDO KAZUNORI
(72)Inventor : ANDO KAZUNORI

(54) ANTI-SPAM METHOD AND ANTI-SPAM PROGRAM

(57)Abstract:
PROBLEM TO BE SOLVED: To provide an anti-spam method and a spam determination program that can reliably detect spam according to a URL address included in the spam.
SOLUTION: The anti-spam method for email sending/receiving comprises a mail analysis step of cracking an email into parts in the smallest unit of mail structure according to separator character strings included in the mail body, a digital fingerprint acquisition step of acquiring a digital fingerprint from every cracked part, a URL information acquisition step of acquiring URL information included in the part, a reference step of referring to a database storing digital fingerprints or URL information acquired/stored from past emails, and a delivery stop step of determining that the email is spam and stopping the delivery of the email if the digital fingerprint acquired in the digital fingerprint acquisition step or the URL information acquired in the URL information acquisition step is stored in the database as a digital fingerprint or URL information related to spam.



(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2005-135024
(P2005-135024A)
(43)公開日　平成17年5月26日(2005.5.26)

(51)Int.Cl.[7]	F I	テーマコード（参考）
G06F 13/00	G06F 13/00　610Q	

審査請求 未請求　請求項の数 12　OL　（全 22 頁）

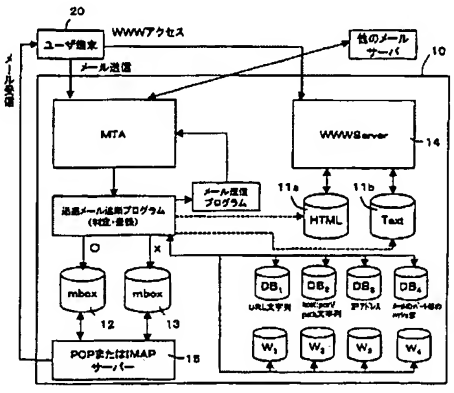
(21)出願番号　特願2003-367895 (P2003-367895)
(22)出願日　平成15年10月28日 (2003.10.28)

（特許庁注：以下のものは登録商標）
1．ＵＮＩＸ

(71)出願人　503396088
安藤　一憲
東京都中央区新富１丁目６－１０　メゾン・ド・ヴィレ銀座東９０６
(74)代理人　100088580
弁理士　秋山　敦
(74)代理人　100111109
弁理士　城田　百合子
(72)発明者　安藤　一憲
東京都中央区新富１丁目６－１０　メゾン・ド・ヴィレ銀座東９０６

(54)【発明の名称】迷惑メール遮断方法及び迷惑メール遮断プログラム

(57)【要約】
【課題】　本発明の目的は、迷惑メールに含まれるＵＲＬアドレスに基づいて、確実に迷惑メールを検出することを可能とした、迷惑メール遮断方法及び迷惑メール判定プログラムを提供する。
【解決手段】　電子メールの送受信における迷惑メール遮断方法であって、電子メールを、メール本文に含まれるセパレータ文字列に従って、メール構造の最小単位であるパートに分解するメール解析工程と、分解したパート毎に電子指紋を取得する電子指紋取得工程と、パートに含まれるＵＲＬ情報を取得するＵＲＬ情報取得工程と、過去の電子メールから取得・蓄積された電子指紋またはＵＲＬ情報が格納されたデータベースを参照する参照工程と、電子指紋取得工程で取得された電子指紋またはＵＲＬ情報取得工程で取得されたＵＲＬ情報が、データベースに迷惑メールに関連する電子指紋またはＵＲＬ情報として格納されていた場合に、電子メールを迷惑メールと判定して該電子メールの配信を停止する配信停止工程と、を備えている。
【選択図】　図２



【特許請求の範囲】
【請求項1】
電子メールの送受信における迷惑メール遮断方法であって、
　前記電子メールを、メール本文に含まれるセパレータ文字列に従って、メール構造の最小単位であるパートに分解するメール解析工程と、
　前記分解したパート毎に電子指紋を取得する電子指紋取得工程と、
　前記パートに含まれるＵＲＬ情報を取得するＵＲＬ情報取得工程と、
　過去の電子メールから取得・蓄積された電子指紋またはＵＲＬ情報が格納されたデータベースを参照するデータベース参照工程と、
　前記電子指紋取得工程で取得された電子指紋または前記ＵＲＬ情報取得工程で取得されたＵＲＬ情報が、前記データベースに迷惑メールに関連する電子指紋またはＵＲＬ情報として格納されていた場合に、前記電子メールを迷惑メールと判定して該電子メールの配信を停止する配信停止工程と、を備えたことを特徴とする迷惑メール遮断方法。
【請求項2】
前記ＵＲＬ情報取得工程では、符号化されたＵＲＬ文字列を復号する処理、復号されたＵＲＬ文字列を分解する処理、前記分解された部分に基づいてホスト名，ポート番号，パス名からなる文字列を生成する処理、前記ホスト名に基づいてＩＰアドレスを取得する処理がなされることを特徴とする請求項１記載の迷惑メール遮断方法。
【請求項3】
電子メールの送受信における迷惑メール遮断方法であって、
　前記電子メールを、メール本文に含まれるセパレータ文字列に従って、メール構造の最小単位であるパートに分解するメール解析工程と、
　前記分解したパート毎に電子指紋を取得する電子指紋取得工程と、
　前記パートに含まれるＵＲＬ情報を取得するＵＲＬ情報取得工程と、
　過去の電子メールから取得・蓄積された電子指紋またはＵＲＬ情報が格納されたデータベースを参照するデータベース参照工程と、
　前記電子指紋取得工程で取得された電子指紋または前記ＵＲＬ情報取得工程で取得されたＵＲＬ情報が既に存在した場合には、データを更新するデータ更新処理を行い、前記電子指紋取得工程で取得された電子指紋または前記ＵＲＬ情報取得工程で取得されたＵＲＬ情報が存在しなかった場合には、新規データとして登録する新規データ登録処理を行うことを特徴とする迷惑メール遮断方法。
【請求項4】
前記データ更新処理または新規データ登録処理では、前記電子指紋またはＵＲＬ情報の重複登録回数、最終更新日時、最初に登場した電子メールのアーカイブ名が登録されることを特徴とする請求項３記載の迷惑メール遮断方法。
【請求項5】
前記最終更新日時から現時点までの期間に応じて、登録の古い順からデータの削除を行うことを特徴とする請求項４記載の迷惑メール遮断方法。
【請求項6】
前記電子メール自体の情報と、前記メール構造に関する情報と、前記電子指紋に関する情報と、前記ＵＲＬ情報と、のうち少なくとも一つをＨＴＭＬファイルまたはテキストファイルとして出力する工程と、
　該ＨＴＭＬファイルまたはテキストファイルをウェブサーバを介して外部に表示する工程と、を備えたことを特徴とする請求項１または３記載の迷惑メール遮断方法。
【請求項7】
前記ＩＰアドレスをＤＮＳサーバに適用可能なブラックリストとして出力する処理がなされることを特徴とする請求項２記載の迷惑メール遮断方法。
【請求項8】
受信した電子メールのヘッダ情報を取得するステップと、

前記電子メールの本文のハッシュ値を計算するステップと、
前記電子メールの本文の復号を行うステップと、
前記電子メールがマルチパート構造の場合に、セパレータ文字列に従って各パートを切り出すステップと、
前記切り出された各パートについて、ヘッダ情報の取得，本文のハッシュ値の計算，本文の復号，各パートの切り出しを繰り返し行うステップと、
前記各パートが可読の場合にＵＲＬ情報を抽出するステップと、
を備えたことを特徴とする迷惑メール遮断プログラム。
【請求項9】
受信した電子メールからＵＲＬ文字列を取得するステップと、
前記ＵＲＬ文字列が符号化されている場合に復号化するステップと、
前記復号化されたＵＲＬ文字列を、スキーム，ユーザ情報，ホスト名，ポート番号，パス名，クエリーに分解するステップと、
前記分解された部分に基づいて、ホスト名，ポート番号，パス名からなる文字列を生成するステップと、
前記ホスト名からＩＰアドレスを取得するステップと、
を備えたことを特徴とする迷惑メール遮断プログラム。
【請求項10】
前記ハッシュ値、前記復号化されたＵＲＬ文字列、前記ホスト名、ポート番号、パス名からなる文字列、前記ＩＰアドレスを迷惑メール判定のための判定情報とし、
新たに受信した電子メールの判定情報と、過去の電子メールから取得・蓄積された判定情報とを対比させるステップと、
前記新たに受信した電子メールの判定情報のうち少なくとも１つが、前記過去の電子メールから取得・蓄積された判定情報に合致した場合に、前記新たに受信した電子メールを迷惑メールとして判定するステップと、を備えたことを特徴とする請求項８または９記載の迷惑メール遮断プログラム。
【請求項11】
前記ハッシュ値、前記復号化されたＵＲＬ文字列、前記ホスト名、ポート番号、パス名からなる文字列、前記ＩＰアドレスを迷惑メール判定のための判定情報とし、
新たに受信した電子メールの判定情報と、過去の電子メールから取得・蓄積された判定情報のデータベースとを対比させるステップと、
新たに受信した電子メールの判定情報のうち少なくとも一つが、過去の電子メールから取得・蓄積された判定情報と合致した場合に、前記データベースのカウンタを１繰り上げるとともに、更新日時を更新して再登録するステップと、
新たに受信した電子メールの判定情報が、過去の電子メールから取得・蓄積された判定情報と合致しない場合に、前記新たに受信した電子メールの判定情報を前記データベースに新規登録するステップと、
を備えたことを特徴とする請求項８または９記載の迷惑メール遮断プログラム。
【請求項12】
前記各ステップにおける処理の結果をＨＴＭＬファイルまたはテキストファイルに保存するステップと、
前記ＨＴＭＬファイルまたはテキストファイルをウェブ上で参照するためのＵＲＬをユーザに送信するステップと、を備えたことを特徴とする請求項８乃至１１いずれか記載の迷惑メール遮断プログラム。

【発明の詳細な説明】
【技術分野】
【0001】
本発明は、迷惑メール遮断方法に係り、特に、メールサーバにおいて迷惑メールのユーザへの配信を遮断する迷惑メール遮断方法と、この迷惑メール遮断方法において実行される迷惑メール遮断プログラムに関する。
【背景技術】

【0002】
近年、インターネット等の通信網を利用した電子メールの送受信において、受信者側の要・不要に係わらず、不特定多数の人に一方的に大量のメールが発信される事態が多発し、社会問題となっている。このようなメールは、受信者側に不快感を与えるとともに、不要なパケット料金の支払い等が発生するものであるので、迷惑メール又はスパムメールと呼ばれている。
【0003】
このような迷惑メールへの対策として、受信者側へ迷惑メールが配信される前に、迷惑メールを自動的に排除・処分するシステムが提案されている（例えば、特許文献１参照）。
【特許文献1】特開２００３－１３１９９９号公報（第３－４頁）
【発明の開示】
【発明が解決しようとする課題】
【0004】
特許文献１に記載された技術は、携帯電話において、メールの本文中にＵＲＬ（Ｕｎｉｆｏｒｍ　Ｒｅｓｏｕｒｃｅ　Ｌｏｃａｔｏｒ）が含まれるか否かを判定し、ＵＲＬが含まれている電子メールを抽出する。ユーザは、ごみ箱メモリに保存された電子メールを参照し、迷惑メールでないものがあった場合には、その電子メールをメール保存メモリに移動させる。このようにして、受信者側に配信された電子メールのなかから、迷惑メールを抽出することを可能としている。
しかし、上記従来技術によれば、送信側でＵＲＬに部分的に改変が加えられてしまった場合には、対応することができないという問題があった。ドメイン名は容易に更新できるため、上記従来技術では迷惑メールの遮断を確実に行うことは不可能であった。
【0005】
迷惑メールを遮断する方法として、上記特許文献１のような技術の他に、レイティング方式と呼ばれる技術がある。レイティング方式とは、インターネット上の各ホームページに対して、「アダルトサイト」、「暴力サイト」などのラベルを付けておき、フィルタリングソフトがそれらのラベルに基づいて、自動的にホームページへのアクセスを制限する方式である。
しかし、上記方法では、レイティングされていないホームページについては対応することができないという問題があった。また、この方法においても、送信側でＵＲＬに部分的に改変が加えられてしまった場合には、対応することができないという問題があった。
【0006】
さらにまた、他の迷惑メールの遮断方法として、メールの本文中に所定の文字を見つけた場合に、そのメールを迷惑メールと判定する方法が知られている。
現在では、一方的に送りつけられるメールに対して、その表題欄に「未承諾広告＊」或いは「！広告！」を付記することが義務づけられている。したがって、メール文中に、「未承諾広告＊」或いは「！広告！」の文字を発見した場合には、迷惑メールとして判定するものである。しかし、上記方法では、言語依存が激しいため、全ての言語で効果を上げるのは非常に難しい。
【0007】
本発明の目的は、迷惑メールに含まれるＵＲＬアドレスに基づいて、確実に迷惑メールを検出することを可能とした、迷惑メール遮断方法及び迷惑メール判定プログラムを提供することにある。
【課題を解決するための手段】
【0008】
前記課題は、請求項１に係る発明によれば、電子メールの送受信における迷惑メール遮断方法であって、前記電子メールを、メール本文に含まれるセパレータ文字列に従って、メール構造の最小単位であるパートに分解するメール解析工程と、前記分解したパート毎に電子指紋を取得する電子指紋取得工程と、前記パートに含まれるＵＲＬ情報を取得する

ＵＲＬ情報取得工程と、過去の電子メールから取得・蓄積された電子指紋またはＵＲＬ情報が格納されたデータベースを参照する参照工程と、前記電子指紋取得工程で取得された電子指紋または前記ＵＲＬ情報取得工程で取得されたＵＲＬ情報が、前記データベースに迷惑メールに関連する電子指紋またはＵＲＬ情報として格納されていた場合に、前記電子メールを迷惑メールと判定して該電子メールの配信を停止する配信停止工程と、を備えたことにより解決される。

【0009】

このように、本発明の迷惑メール遮断方法によれば、電子メールの各パートの電子指紋と、電子メールに含まれるＵＲＬ情報に基づいて、ユーザ宛に繰り返し配送される迷惑メールを判定するようにされている。このため、入れ子構造のパートにＵＲＬ情報が隠蔽されていたり、ＵＲＬ情報が符号化等により偽装されているときにも、ユーザ宛に繰り返し送付される情報を把握し、迷惑メールを確実に遮断することが可能となる。

【0010】

前記ＵＲＬ情報取得工程では、符号化されたＵＲＬ文字列を復号する処理、復号されたＵＲＬ文字列を分解する処理、前記分解された部分に基づいてホスト名，ポート番号，パス名からなる文字列を生成する処理、前記ホスト名に基づいてＩＰアドレスを取得する処理がなされる。

上記処理により、ＵＲＬ情報が符号化により偽装されている場合であっても、復号化することにより偽装を見破ることが可能となる。また、改変することが難しい「ホスト名，ポート番号，パス名からなる文字列」を生成したり、ＩＰアドレスを取得することにより、繰り返し送付される情報を確実に発見することが可能となる。

【0011】

また、本発明におけるデータベースの構築は、請求項３に記載のように行う。

すなわち、電子メールの送受信における迷惑メール遮断方法であって、前記電子メールを、メール本文に含まれるセパレータ文字列に従って、メール構造の最小単位であるパートに分解するメール解析工程と、前記分解したパート毎に電子指紋を取得する電子指紋取得工程と、前記パートに含まれるＵＲＬ情報を取得するＵＲＬ情報取得工程と、過去の電子メールから取得・蓄積された電子指紋またはＵＲＬ情報が格納されたデータベースを参照するデータベース参照工程と、前記電子指紋取得工程で取得された電子指紋または前記ＵＲＬ情報取得工程で取得されたＵＲＬ情報が既に存在した場合には、データを更新するデータ更新処理を行い、前記電子指紋取得工程で取得された電子指紋または前記ＵＲＬ情報取得工程で取得されたＵＲＬ情報が存在しなかった場合には、新規データとして登録する新規データ登録処理を行う。

【0012】

前記データ更新処理または新規データ登録処理では、前記電子指紋またはＵＲＬ情報の重複登録回数、最終更新日時、最初に登場した電子メールのアーカイブ名が登録される。そして、前記最終更新日時から現時点までの期間に応じて、登録の古い順からデータの削除が行われる。

【0013】

また、前記電子メール自体の情報と、前記メール構造に関する情報と、前記電子指紋に関する情報と、前記ＵＲＬ情報のうち少なくとも一つをＨＴＭＬファイルまたはテキストファイルとして出力する工程と、該ＨＴＭＬファイルまたはテキストファイルをウェブサーバを介して外部に表示する工程とを備えた構成とすると、外部からでも解析結果を参照することが可能となる。

【0014】

また、前記ＩＰアドレスをＤＮＳサーバに適用可能なブラックリストとして出力する処理がなされると、このブラックリストをＤＮＳ経由で参照することで、リモートホストでも迷惑メールの遮断が可能となる。

【0015】

本発明の迷惑メール遮断プログラムは、電子メールの解析と、ＵＲＬ情報の抽出を行う

ために、次の各ステップを備えている。

すなわち、受信した電子メールのヘッダ情報を取得するステップと、前記電子メールの本文のハッシュ値を計算するステップと、前記電子メールの本文の復号を行うステップと、前記電子メールがマルチパート構造の場合に、セパレータ文字列に従って各パートを切り出すステップと、前記切り出された各パートについて、ヘッダ情報の取得，本文のハッシュ値の計算，本文の復号，各パートの切り出しを繰り返し行うステップと、前記各パートが可読の場合にＵＲＬ情報を抽出するステップと、を備えている。

【0016】

また、本発明の迷惑メール遮断プログラムは、ＵＲＬ情報の解析を行うために、次の各ステップを備えている。

すなわち、受信した電子メールからＵＲＬ文字列を取得するステップと、前記ＵＲＬ文字列が符号化されている場合に復号化するステップと、前記復号化されたＵＲＬ文字列を、スキーム，ユーザ情報，ホスト名，ポート番号，パス名，クエリーに分解するステップと、前記分解された部分に基づいて、ホスト名，ポート番号，パス名からなる文字列を生成するステップと、前記ホスト名からＩＰアドレスを取得するステップと、を備えている。

【0017】

さらに、本発明の迷惑メール遮断プログラムは、受信した電子メールを迷惑メールとして判定するために、次の各ステップを備えている。

すなわち、前記ハッシュ値、前記復号化されたＵＲＬ文字列、前記ホスト名、ポート番号、パス名からなる文字列、前記ＩＰアドレスを迷惑メール判定のための判定情報とし、新たに受信した電子メールの判定情報と、過去の電子メールから取得・蓄積された判定情報とを対比させるステップと、前記新たに受信した電子メールの判定情報のうち少なくとも１つが、前記過去の電子メールから取得・蓄積された判定情報に合致した場合に、前記新たに受信した電子メールを迷惑メールとして判定するステップを備えている。

【0018】

さらにまた、本発明の迷惑メール遮断プログラムは、データベースを構築するために、次の各ステップを備えている。

すなわち、前記ハッシュ値、前記復号化されたＵＲＬ文字列、前記ホスト名、ポート番号、パス名からなる文字列、前記ＩＰアドレスを迷惑メール判定のための判定情報とし、新たに受信した電子メールの判定情報と、過去の電子メールから取得・蓄積された判定情報のデータベースとを対比させるステップと、新たに受信した電子メールの判定情報のうち少なくとも一つが、過去の電子メールから取得・蓄積された判定情報と合致した場合に、前記データベースのカウンタを１繰り上げるとともに、更新日時を更新して再登録するステップと、新たに受信した電子メールの判定情報が、過去の電子メールから取得・蓄積された判定情報と合致しない場合に、前記新たに受信した電子メールの判定情報を前記データベースに新規登録するステップと、を備えている。

【0019】

さらにまた、本発明の迷惑メール遮断プログラムは、解析結果をユーザへ報告するために、次の各ステップを備えている。

すなわち、前記各ステップにおける処理の結果をＨＴＭＬファイルまたはテキストファイルに保存するステップと、前記ＨＴＭＬファイルまたはテキストファイルをウェブ上で参照するためのＵＲＬをユーザに送信するステップと、を備えている。

【発明の効果】

【0020】

本発明の迷惑メール遮断方法及び迷惑メール遮断プログラムによれば、マルチパート構造の電子メールの中にＵＲＬ情報が隠蔽されている場合や、ＵＲＬ情報が偽装されている場合であっても、電子メールの解析及びＵＲＬ情報の解析により、これらの隠蔽や偽装を見破り、迷惑メールを確実に検出することが可能となる。

【発明を実施するための最良の形態】

【0021】
以下、本発明の一実施の形態を図面に基づいて説明する。なお、以下に説明する部材，配置等は本発明を限定するものでなく、本発明の趣旨の範囲内で種々改変することができるものである。
【0022】
図１は一般的な電子メール送受信のシステム構成を示す説明図、図２は本発明の実施の形態における迷惑メール遮断方法及びプログラムが実行されるシステムの構成図、図３は迷惑メール検出の手順を示すブロック図、図４はメール受信からメールボックス格納までの流れを示す説明図、図５はメール受信から解析結果提示までの流れを示す説明図、図６はプログラムのオプションの例を示す一覧、図７は実際のａｌｉａｓの例を示す一覧、図８乃至図１０はＭＩＭＥメールの構造を示す説明図、図１１はマルチパートの電子メールの復号化処理を示すフローチャート、図１２はＵＲＬの抽出・解析処理を示すフローチャート、図１３は符号化された文字の一覧表、図１４は符号化されたＵＲＬと復号化されたＵＲＬの例を示す説明図、図１５は迷惑メールの判定処理を示すフローチャート、図１６はデータベースへの登録処理を示すフローチャート、図１７は解析結果をユーザに返信する処理を示すフローチャート、図１８及び図１９は解析結果の一例を示す説明図、図２０は解析結果を保存する記憶部の構造を示す説明図、図２１はＵＲＬに使用可能な文字の具体例を示す説明図である。
【0023】
図１に、一般的な電子メール送受信のシステム構成を示す。図示されているように、電子メール送受信のシステムは、メールサーバ１０と、ユーザ端末２０と、インターネット３０等の通信回線網とを備えて構成されている。インターネット３０には、複数のメールサーバ２０が接続されており、メールサーバ２０間でメールの送受信処理が行われる。
ユーザ端末２０は、メールを作成及び送受信する機能を持つメールソフトウェアを備えており、ユーザ端末２０がメールサーバ１０に接続された状態において、ユーザ端末２０はメールサーバ１０のメールボックスから、ユーザ宛のメールを取得することができる。なお、ユーザ端末２０が携帯電話である場合、メールサーバ１０とユーザ端末２０とは電通信回線を介して接続される。
【0024】
図２は、本発明の実施の形態における迷惑メール遮断方法及びプログラムが実行されるシステムの構成図である。本システムは、本発明の実施の形態における迷惑メール遮断方法を実行するサーバ装置としてメールサーバ１０を備えている。メールサーバ１０には、迷惑メール遮断プログラムや、ＭＴＡ（Ｍａｉｌ　Ｔｒａｎｓｆｅｒ　Ａｇｅｎｔ）を含むメールサーバソフトウェアがインストールされている。
メールサーバ１０は、例えばワークステーションやパーソナルコンピュータ（パソコン）をはじめとするコンピュータ装置により構成される。メールサーバ１０は、電子指紋として取得されるメールのハッシュ値や、メールに含まれていたＵＲＬ情報が格納されるデータベースDB_1～DB_4、ホワイトリスト情報が格納されるデータベースW_1～W_4、メールの解析結果が格納されるデータ格納部１１ａ，１１ｂ、メールボックス１２、迷惑メールボックス１３、ウェブサーバー１４、ＰＯＰサーバまたはＩＭＡＰサーバ１５を備えて構成されている。また、メールサーバ１０は、ＭＴＡや各種プログラムに従って各種処理を行う制御部（図示せず）を備えている。
【0025】
図３に示すように、本例のシステムでは、DB_1～DB_4への登録処理として、標準入力から電子メールを読み込む処理Ｓ１と、電子メールの各パートの要素に対する処理Ｓ２と、ＤＢ参照によるＤＢへの登録処理Ｓ３と、解析結果を登録されたユーザに返信する処理Ｓ４と、が行われる。さらに、迷惑メールの判定処理として、標準入力から電子メールを読み込む処理Ｓ１と、電子メールの各パートの要素に対する処理Ｓ２と、ＤＢ参照による判定処理Ｓ５と、迷惑メールの遮断処理Ｓ６とが行われる。
【0026】

図４に示すように、メールサーバ１０は、ユーザ端末２０から発信された電子メール、または他のメールサーバから転送されてきた電子メールを受信する。電子メールは、図示しない電子メール保存部に格納されるが、電子メールそのものを受信する前に、受信された電子メールの宛先の判定を行い、その電子メールの宛先がローカルであるかリモートであるかの判定を行う。宛先がローカルである場合には、その電子メールについてローカル処理がなされる。宛先がリモートの場合には、その電子メールは他のサーバへ転送される。

ローカル処理がなされる場合、電子メールは、判定プログラムに入力され、迷惑メールであるか否かが判定される。迷惑メールではないと判定された電子メールは、宛先が指定され、ローカルメーラープログラムによりメールボックス１２へ配信される。

【0027】

DB_1～DB_4には、迷惑メールを判定するために必要な各種データが格納されている。

DB_1～DB_4には、電子メールから抽出された情報が格納される。DB_1～DB_4に格納される情報としては、電子メールに含まれるＵＲＬ文字列（復号化されたもの）、ＵＲＬから容易に可変できる部分を除いた”ｈｏｓｔ：ｐｏｒｔ／ｐａｔｈ”の文字列、ＵＲＬのホストをＤＮＳ検索して得られるＩＰアドレス、メールのパートごとのハッシュ値、の４種類がある。メールのパートごとのハッシュ値は、電子指紋として取得されるものである。

【0028】

DB_1～DB_4に対応して、ホワイトリスト登録用のデータベースW_1～W_4が設けられている。ホワイトリスト登録用のデータベースW_1～W_4には、迷惑メール関連でないことが予め判っている情報が格納される。

W_1～W_4に登録される情報はDB_1～DB_4に格納される情報と対になっており、復号後のＵＲＬ文字列、”ｈｏｓｔ：ｐｏｒｔ／ｐａｔｈ”の文字列、ＩＰアドレス、メールのパートごとのハッシュ値、が格納される。ここに記録された各情報に該当したメールについては、迷惑メール判定の対象外とされる。

DB_1～DB_4及びW_1～W_4はハッシュ形式であり、「復号後のＵＲＬ文字列」、「”ｈｏｓｔ：ｐｏｒｔ／ｐａｔｈ”の文字列」、「ＩＰアドレス」、「メールのパートごとのハッシュ値」、がキーとされ、「重複登録回数」、「最終更新日時（ｕｎｉｘ　ｔｉｍｅ）」、「最初に登場したメールのアーカイブ名」がデータとして登録される。

【0029】

なお、ＩＰアドレスに関しては、別途、ＤＮＳＢＬ用のデータ形式でファイルへ出力される。このようにすると、取得されたＩＰアドレスをいわゆるＤＮＳＢＬの仕組みで参照できるようになり、リモートホストでも登録されたＩＰアドレスをベースにした迷惑メール遮断が可能とされる。最近では、ＲＦＣ２１３６を用いて動的にＤＮＳの更新を行うことも可能になり、この技術を利用する場合は、データ形式のファイルが不要となる。

【0030】

本例では、図５に示すように、DB_1～DB_4またはW_1～W_4へのデータの登録・削除の動作時に、ユーザ宛に解析結果レポートを送信することが可能である。メールサーバ１０には、ＵＲＬ情報の解析結果が格納されるデータ格納部１１ａ，１１ｂが設けられている。

ユーザ提示される情報は、解析結果をＨＴＭＬファイルにすることにより形成される。ＨＴＭＬ出力されたファイルは、データ格納部１１ａに格納される。

また、電子メールそのものがテキストファイルとして出力される。この情報は、データ格納部１１ｂに格納される。データ格納部１１ａ，１１ｂへそれぞれファイルを格納するときは、最初に登場したメールのアーカイブ名がファイル名とされる。電子メールのテキストファイルは、解析結果のページからリンクを張って参照可能とされる。

データ格納部１１ａ及び１１ｂに格納された解析結果は、ウェブサーバ１４を介して、外部から閲覧することができる。閲覧を希望するユーザには、これらの情報を閲覧するた

めのＵＲＬがメールにより送信される。

ユーザは解析結果を参照し、ホワイトリストに登録すべきものがあるかどうか検討する。ホワイトリストに登録するものがあれば、後述するように、そのＵＲＬ情報をホワイトリスト登録用のアドレスにメール送信する。

【0031】

メールサーバ１０では、DB_1～DB_4、W_1～W_4へのデータの登録及び削除や、データ格納部１１ａ，１１ｂへのデータ出力において、ａｌｉａｓ（別名指定）を利用したプログラムの呼び出しを行っている。

この場合、特定のメールアドレスにメールを転送することにより、ａｌｉａｓファイルに書かれたプログラムが所定のオプション付きで起動され、転送されたメールがそのプログラムに入力される、という動作がＭＴＡにより行われる。

図６に、本例におけるオプションの一例を示す。また、図７に、メールサーバ１０のａｌｉａｓファイルに書かれているプログラムの一例を示す。

例えば、迷惑メールに含まれるＵＲＬ情報等を、データベースDB_1～DB_4へ登録する場合は、登録用のメールアドレス（－ｂオプションの指定されているａｌｉａｓ）に対して電子メールを転送することにより行う。

また、特定の情報をデータベースDB_1～DB_4から削除する場合は、その情報をメールに書いて、削除用のメールアドレス（－ｅオプションの指定されているａｌｉａｓ）に送信するだけで良い。

さらに、ホワイトリストデータベースW_1～W_4への登録であれば、ホワイトリスト登録用のメールアドレス（－ｗオプションの指定されているａｌｉａｓ）に対して、本文に登録する情報を書いてメールを送信する。

本例のシステムでは、迷惑メールを判定するための情報として、送信元情報ではなく、誘導先情報であるＵＲＬ情報と、メールの各パートのハッシュ値を用いているので、結果として、上記のように単純なメール送信によるＤＢへの登録が可能となっている。

【0032】

DB_1～DB_4は、定期的にメンテナンスが行われ、ＤＢの更新プログラムが定期的に実行される。ＤＢの更新プログラムは、現在日時と、DB_1～DB_4に記録された更新日時を比較し、所定期間以上経過したデータは削除する。例えば、「ＩＰアドレス」は３ヶ月、「メールの各パートのハッシュ値」は３ヶ月、「復号後のＵＲＬ文字列」は１ヶ月、「”ｈｏｓｔ：ｐｏｒｔ／ｐａｔｈ”の文字列」は２ヶ月で削除される。

【0033】

「メールの各パートのハッシュ値」、「復号後のＵＲＬ文字列」、「ＩＰアドレス」、「”ｈｏｓｔ：ｐｏｒｔ／ｐａｔｈ”の文字列」は、次のようにして取得される。

「メールの各パートのハッシュ値」は、電子メールがＭＩＭＥ（Ｍｕｌｔｉｐｕｒｐｏｓｅ　Ｉｎｔｅｒｎｅｔ　Ｍａｉｌ　Ｅｘｔｅｎｓｉｏｎｓ）で規定されたパートからなるメールとして送付されたときに、その各パートから得られるものである。

ハッシュ値とは、任意の長さのデータを固定長のデータに投影するハッシュ関数を用いて計算される固定長のデータを指すものである。ハッシュ関数の種類によって、得られるハッシュ値のデータ長は異なる。

本例では、１６０ｂｉｔのハッシュ値が得られるＳＨＡ１と称するハッシュ関数を用いている。ハッシュ関数としては、上記ＳＨＡ１の他に、ＭＤ５（１２８ｂｉｔ）、ＲＩＰＥＭＤ（１６０ｂｉｔ）、ＳＨＡ２５６（２５６ｂｉｔ）などがあるが、本例では、実用上十分な精度があり、計算量も適切なＳＨＡ１を用いている。

【0034】

ハッシュ関数を使った場合、データの内容が１バイトでも異なると、全く別のハッシュ値が生成される。このため、ハッシュ値が同一であれば同一のデータであることが判明し、ハッシュ値が異なれば別の内容であると判定することができる。

ハッシュ値を利用することにより、同一のデータが繰り返し送付された場合に、それを検出することができ、迷惑メールとして判定することが可能となる。

【0035】
ＭＩＭＥメールがＭＩＭＥ－ｍｕｌｔｉｐａｒｔである場合、パートを入れ子にすることが可能であるため、解析は再帰的に行う。
ここで、ＭＩＭＥメールの構造について説明する。
ＭＩＭＥメールの最小単位になるパートは、図８に示すように、ヘッダ部、空行、本文から構成されているが、図９及び図１０に示すようにマルチパートの構造からなるものがある。この場合、ヘッダ部には、例えば「Ｃｏｎｔｅｎｔ－Ｔｙｐｅ：ｍｕｌｔｉｐａｒｔ／ｍｉｘｅｄ」のように記載されている。
マルチパートの構造は、セパレータ文字列を区切りにして、パートの中に、さらにパートを備えた構造となっている。
【0036】
図９及び図１０は、入れ子構造のマルチパートとされた例であり、図９の例では、本文の中にさらにパートが一つ設けられている。図１０の例では、本文の中にパートが一つ設けられ、このパートの中に、さらにパートが設けられている。セパレータ文字列の前後にマイナス記号を２つずつ付加したものがマルチパート終了を意味する。
また、入れ子構造の他に、パートの中に複数のパートが並列に格納されている構造や、並列に格納された構造と、入れ子構造を組み合わせた構造としたものもある。
迷惑メールでは、上記各パートにＵＲＬを含ませて、ＵＲＬを隠蔽していることがある。このため、各パートをＭＩＭＥ－ｍｕｌｔｉｐａｒｔの構造に沿って復号化し、さらに各パートに含まれるＵＲＬを抽出する必要がある。
【0037】
本例のメールサーバ１０では、プログラムにしたがって、マルチパートの電子メールの解析が行われる。図１１は、マルチパートの電子メールを、その構造に沿って復号化する処理の流れを示すものである。この処理は、図３のＳ２の処理（パートの要素に対する処理）に該当する。
最初に、本文とヘッダからなるメールが、解析対象のパートとして渡される（ステップＳ１１）。
次に、パートのヘッダ部の解析が行われる（ステップＳ１２）。ヘッダ部には、符号化方式、コンテンツの種類等の情報が記載されている。
ヘッダ部に、例えば「Ｃｏｎｔｅｎｔ－Ｔｙｐｅ：ｍｕｌｔｉｐａｒｔ／ｍｉｘｅｄ」と記載されている場合は、電子メールがマルチパートの構造であると判断される。電子メールがマルチパートの構造である場合は、セパレータ文字列を読み取る。
【0038】
ステップＳ１３では、本文のハッシュ計算を行う。
さらに、各パートは、ｂａｓｅ６４やｑｕｏｔｅｄ－ｐｒｉｎｔａｂｌｅという手法で符号化されていることがあるため、ステップＳ１４で、ヘッダ部に符号化指定が記載されているかどうか判定する（ステップＳ１４）。符号化指定がある場合は（ステップＳ１４；Ｙｅｓ）、ステップＳ１５で復号化処理を行い、本文の復号化を行う。符号化指定があるのに、符号化されていない場合は、復号化処理は行わず、ステップＳ１６に進む。
また、符号化指定がない場合は（ステップＳ１４；Ｎｏ）、ステップＳ１６に進む。
ステップＳ１６では、ステップＳ１２で解析したヘッダ部の情報に基づき、パートがマルチパートの構造であるか否かが判定される。
マルチパートの構造ではない場合（ステップＳ１６；Ｎｏ）、ステップＳ１７に進み、パートが可読なものであるか否かを判定する。パートのメディアタイプが、ｔｅｘｔ／ｐｌａｉｎ、ｔｅｘｔ／ｈｔｍｌ等可読なものの場合には（ステップＳ１７；Ｙｅｓ）、ステップＳ１８でＵＲＬの抽出・解析処理を行う。
パートのメディアタイプが可読なものでない場合は（ステップＳ１７；Ｎｏ）、処理を終了する。
【0039】
また、ステップＳ１６で、パートがマルチパートの構造であると判定された場合は（ス

テップＳ１６；Ｙｅｓ）、セパレータ文字列に従って、パートの中に入れ込まれているパートを切り出す処理を行う（ステップＳ１９）。そして、切り出された各パートについて、ステップＳ１１～ステップＳ１９の処理を同様に行う。

この処理により、電子メールを構成する各パートについて、それぞれハッシュ値が求められるとともに、各パートにＵＲＬ情報が含まれている場合は、その情報が抽出・解析される。

このように、本例ではＭＩＭＥマルチパートが入れ子構造を許していることに対応して、解析手続きを再帰呼び出しして、完全な解析を実現している。

なお、解析結果を収納する記憶部（メモリ）の構造についても、ＭＩＭＥマルチパートの構造にしたがって入れ子構造になっている。図２０（ａ）は、１つのパートを納める記憶部の構造を示すものである。入れ子構造のパートの場合は、内包するパートを２５６個まで格納できる。内包されるパートについては、実際には図２０（ａ），（ｂ）に示すように、この定義で示されている構造体へのポインタとして格納される。

【0040】

次に、図１２に基づいて、ＵＲＬの抽出・解析処理（図１１の処理のステップＳ１８）について説明する。

ステップＳ２１では、電子メールに所定のスキームが含まれているかどうかのサーチがなされる。

ステップＳ２１のスキームのサーチにおいて、ターゲットとなるＵＲＬは、以下の４種類の仕様（スキーム）に対応するＵＲＬである。

「ｈｔｔｐ：／／」で始まるＵＲＬ。このＵＲＬは、ＨＴＴＰ（Ｈｙｐｅｒｔｅｘｔ Ｔｒａｎｓｆｅｒ Ｐｒｏｔｏｃｏｌ）、すなわちウェブサーバとウェブクライアントの間でＨＴＭＬ文書を送受信するための通信プロトコルに対応しているものである。

「ｈｔｔｐｓ：／／」で始まるＵＲＬ。このＵＲＬは、ＨＴＴＰＳ（Ｈｙｐｅｒｔｅｘｔ Ｔｒａｎｓｆｅｒ Ｐｒｏｔｏｃｏｌ Ｓｅｃｕｒｉｔｙ）、すなわち、ＨＴＴＰとＳＳＬ（Ｓｅｃｕｒｅ Ｓｏｃｋｅｔｓ Ｌａｙｅｒ）の暗号化機能を組み合わせた通信プロトコルに対応するものである。

「ｒｔｓｐ：／／」で始まるＵＲＬ。このＵＲＬは、ＲＴＳＰ（Ｒｅａｌ Ｔｉｍｅ Ｓｔｒｅａｍｉｎｇ Ｐｒｏｔｏｃｏｌ）、すなわち、オーディオ・データやビデオ・データを実時間転送するための通信プロトコルに対応するものである。

「ｆｔｐ：／／」で始まるＵＲＬ。このＵＲＬは、ＦＴＰ（Ｆｉｌｅ Ｔｒａｎｓｆｅｒ Ｐｒｏｔｏｃｏｌ）、すなわち、ファイル転送プロトコルに対応するものである。

ここで、ＵＲＬが１つもない場合には、以下のステップにおけるデータは生成されず、ハッシュ値だけの参照となる。

【0041】

ステップＳ２２では、ＵＲＬの終点が確定される。

そして、ステップＳ２３では、ＵＲＬが符号化されているかどうかの判定がなされる。符号化されている場合（ステップＳ２３；Ｙｅｓ）、ステップＳ２４で復号化が行われ、再度、ＵＲＬの終点が確定される。

ステップＳ２３及びステップＳ２４の処理は、符号化により偽装されたＵＲＬに対応するために行われるものであり、この処理により、「復号後のＵＲＬ文字列」を取得することができる。

符号化の手法として、例えば文字実体参照、数値実体参照、エスケープ符号化がある。

文字実体参照は、ＤＴＤ（Ｄｏｃｕｍｅｎｔ Ｔｙｐｅ Ｄｅｆｉｎｉｔｉｏｎ；文書型定義）で定義された名前で文字を指定する手法であり、文字コード位置が「＆」と「；」で囲まれる記載となる。

文字実体参照の例として、例えば「＆」であれば、「＆ａｍｐ；」と表示される。また、例えば「¥」であれば、「＆ｙｅｎ；」と表示される。また、例えば「＜」であれば、「＆ｌｔ；」と表示される。

数値実体参照において、例えば１０進数で指定する場合は、文字コード位置が「＆＃」

と「；」で囲まれる記載となる。
数値文字参照の例として、例えば「＆」であれば「＆＃３８；」と表示される。また、例えば「〒」であれば、「＆＃１６５；」と表示される。また、例えば「＜」であれば、「＆＃６０；」と表示される。
【0042】
エスケープ符号化も、本来ＵＲＬに使用できない文字を取り込むのに使用される枠組みであり、ＵＲＬの偽装に使用されることがある。エスケープ符号化は、ＲＦＣ２３９６に記載されているように、三連文字として符号化されるものである。この三連文字は、「％」文字の後に、２つの１６進数字が続く形で構成される。ＵＲＬにおいてエスケープ符号化がなされた場合は、「ｗ」が「％７７」と表示される。
このように、符号化に際しては特定の文字が使用されているため、ステップＳ２３では文字が符号化されているか否かを判定するため、ＵＲＬのなかに、「＆」、「＃」、「；」、「％」の文字が使用されているか否かがチェックされる。
【0043】
ステップＳ２４では、符号化された文字を復号するために、文字一覧表が読み込まれる。文字一覧表は、例えば図１３に示すように、符号化された文字と、復号した文字とが対になって登録されている。そして、この一覧表を参照して、符号化された文字の復号がなされる。
図１４に、復号化の一例を示す。図において、上段が符号化されたＵＲＬ、下段が復号化されたＵＲＬである。例１は数値実体参照による符号化の例、例２は数値実体参照（セミコロンなし）による符号化の例、例３はエスケープ（ＵＲＩ－ｅｎｃｏｄｅ）による符号化の例である。
【0044】
ここで、ＵＲＩに使用可能な文字で余計な部分を切り捨てる。つまり、抽出されたＵＲＬの各部にそれぞれ使用可能な文字のチェックをかけ、使用できない文字が発見されるとそこを末尾と判定する。図２１に、末尾判定に使用される文字、ユーザ情報部分に使用可能な文字、ユーザ情報以外の部分に使用可能な文字の具体例を示す。
ステップＳ２５では、ＵＲＬを要素に分解する処理がなされる。
ＵＲＬはインターネットで使用される様々なリソースの場所を表すものであり、ＵＲＬにはリソースを取り出すためのプロトコルやディレクトリ、ポートなどの情報が含まれている。
ＵＲＬは、一般に次のような形式とされている。
「ｓｃｈｅｍｅ：／／ｕｓｅｒｉｎｆｏ＠ｈｏｓｔ：ｐｏｒｔ／ｐａｔｈ？ｑｕｅｒｙ」
なお、「ｕｓｅｒｉｎｆｏ＠」、「：ｐｏｒｔ」、「ｐａｔｈ」、「？ｑｕｅｒｙ」は省略されていることもある。
ステップＳ２５では、ＵＲＬを、「ｓｃｈｅｍｅ」、「ｕｓｅｒｉｎｆｏ」、「ｈｏｓｔ」、「ｐｏｒｔ」、「ｐａｔｈ」、「ｑｕｅｒｙ」に分解する処理がなされる。
【0045】
具体的な例を挙げると、例えば、
「ｈｔｔｐ：／／ａｎｄｏ：ｐａｓｓｗｏｒｄ＠ｎｓ＠ｗｗｗ．ｐｐｍｌ．ｔｖ／」
というＵＲＬの場合は、次のように分解される。
「ｕｓｅｒｉｎｆｏ」は「ａｎｄｏ：ｐａｓｓｗｏｒｄ＠ｎｓ」。
「ｈｏｓｔ」は「ｗｗｗ．ｐｐｍｌ．ｔｖ」。
「ｐｏｒｔ」は省略されているが、ｈｔｔｐなので「８０」。ポート番号は、ｈｔｔｐなら８０、ｈｔｔｐｓなら４４３、ｆｔｐなら２１、ｒｔｓｐなら５５４、が適用される。
「ｐａｔｈ」は「／」。
「ｑｕｅｒｙ」は「なし」。
「ｓｃｈｅｍｅ」は「ｈｔｔｐ」。
【0046】

迷惑メールに記載されるＵＲＬは、特定のＵＲＬとして拾われることを防止するため、文字等を追加して異なるＵＲＬに見せかけているものがある。

例えば、ｕｓｅｒｉｎｆｏにわざと「＠」を含む文字列を使って、「ｓｃｈｅｍｅ：／／ｉｎｔｒｕｄｅ＠ｉｎｔｅｒｃｅｐｔ＠ｈｏｓｔ：ｐｏｒｔ／」のようにされているＵＲＬがある。この場合は、「ｉｎｔｒｕｄｅ＠ｉｎｔｅｒｃｅｐｔ」がｕｓｅｒｉｎｆｏとして扱われるべき文字列となる。

また、ｕｓｅｒｉｎｆｏに「空白文字」を含ませて、「ｈｔｔｐ：／／ｕ　ｓ　ｅ　ｒ　ｉ　ｎ　ｆ　ｏ＠ｈｏｓｔ：ｐｏｒｔ／」のようにされているＵＲＬがある。この場合は、「ｕ　ｓ　ｅ　ｒ　ｉ　ｎ．ｆ　ｏ」がｕｓｅｒｉｎｆｏとして扱われるべき文字列となる。

さらに、ｕｓｅｒｉｎｆｏに「改行」を含ませて、「ｈｔｔｐ：／／ｕｓｅｒ（改行）ｉｎｆｏ＠ｓｔｒｉｎｇ＠ｈｏｓｔｎａｍｅ：ｐｏｒｔ／」のようにされているＵＲＬがある。この場合は、「改行」を除き、「ｕｓｅｒｉｎｆｏ＠ｓｔｒｉｎｇ」がｕｓｅｒｉｎｆｏとして扱われるべき文字列となる。

【0047】

また、ホスト名部分において偽装されていることがある。

ａ．「ｗｗｗ．ｐｐｍｌ．ｔｖ」を、大文字と小文字を入れ替えることにより「ｗｗｗ．ＰｐＭｌ．ｔｖ」としている。

ｂ．ホスト名である「ｗｗｗ．ｐｐｍｌ．ｔｖ」を、ＩＰアドレスである「２１０．１３８．３５．２７」としている。

ｃ．ＩＰアドレス「２１０．１３８．３５．２７」を、Ｈｅｘａｄｅｃｉｍａｌ（１６進数）形式を用いて「０ｘＤ２．０ｘ８Ａ．０ｘ２３．０ｘ１Ｂ」としている。

【0048】

ｄ．ＩＰアドレス「２１０．１３８．３５．２７」を、Ｏｃｔａｌ（８進数）形式を用いて「０３２２．０２１２．０４３．０３３」としている。

ｅ．ＩＰアドレス「２１０．１３８．３５．２７」を、Ｈｅｘａｄｅｃｉｍａｌ形式（ｕｎｓｉｇｎｅｄ　ｌｏｎｇ；符号なし長整数）を用いて「０ｘＤ２８Ａ２３１Ｂ」としている。

ｆ．ＩＰアドレス「２１０．１３８．３５．２７」を、Ｄｅｃｉｍａｌ（１０進数）形式（ｕｎｓｉｇｎｅｄ　ｌｏｎｇ）を用いて「３５３２２６８３１５」としている。

【0049】

上記ａのように、「ｗｗｗ．ＰｐＭｌ．ｔｖ」のように大文字で記載されたものについては、「ｗｗｗ．ｐｐｍｌ．ｔｖ」のように小文字に変換する。

また、上記ｂ～ｆのように、ＩＰアドレスを別の形式で書き換えてあるものについて、「２１０．１３８．３５．２７」のようにドットで区切られた１０進数での表記に統一する。

【0050】

ステップＳ２６では、ステップＳ２５での分解結果から、「ｈｏｓｔ：ｐｏｒｔ／ｐａｔｈ」文字列が生成される。この文字列は、ＵＲＬの要素のなかで容易に可変できない部分であり、迷惑メールの判定を行う際に有効に用いることができる。

「ｈｔｔｐ：／／ａｎｄｏ：ｐａｓｓｗｏｒｄ＠ｎｓ＠ｗｗｗ．ｐｐｍｌ．ｔｖ／」の例では、「ｈｏｓｔ：ｐｏｒｔ／ｐａｔｈ」文字列として「ｗｗｗ．ｐｐｍｌ．ｔｖ：８０／」が生成される。

【0051】

ステップＳ２７では、ホスト名からＩＰアドレスを取得する処理が行われる。

この処理は、上記ステップＳ２５でＩＰアドレスが得られた場合には省略される。

ステップＳ２８では、ステップＳ２６でホスト名「ｗｗｗ．ｐｐｍｌ．ｔｖ」が得られた場合、このホスト名に基づいてＩＰアドレスが取得される。ＩＰアドレスは、ＤＮＳ（Ｄｏｍｅｉｎ　Ｎａｍｅ　Ｓｙｓｔｅｍ）サーバへのアクセスにより得ることができる。

ＩＰアドレスは得られた数だけ全てが取得される。

【0052】

このようにして、ＵＲＬの抽出及び解析が終了する。ステップＳ２１～ステップＳ２７の処理により、ＵＲＬが様々な形で偽装されていても、その偽装を解いて、「復号後のＵＲＬ文字列」、「”ｈｏｓｔ：ｐｏｒｔ／ｐａｔｈ”の文字列」、「ＩＰアドレス」を得ることができる。

ＵＲＬの解析が終了すると、解析結果は図２０に示す記憶部に保存される。そして、その後、DB_1～DB_4参照による判定処理及びDB_1～DB_4への登録処理等の各処理が行われる。

【0053】

ここで、ＵＲＬの抽出・解析処理のステップＳ２４の後に行う例外処理について説明する。この処理では、リダイレクターの排除が行われる。

迷惑メールに記載されるＵＲＬは、特定のＵＲＬとして禁止されることを防止するため、無関係なサイトのリダイレクト機能を用いていることがある。すなわち、本来のＵＲＬの前に、別のＵＲＬを付加し、この別のＵＲＬにアクセスしてきたユーザを、強制的に本来的に見せたいページへ導くものである。

【0054】

リダイレクターを不正に利用したＵＲＬは、例えば、

「ｈｔｔｐ：／／ｓｒｄ．ａｂｃｄｅ．ｃｏｍ／ｄｒｓｔ／８００５０１３７８２５５／＊ｈｔｔｐ：／ｗｗｗ．３６５ｐｈａｒｍ１．ｃｏｍ／」

のように記載されている。

上記ＵＲＬのうち、「ｈｔｔｐ：／ｗｗｗ．３６５ｐｈａｒｍ１．ｃｏｍ／」が本来的にユーザに見せたいページを示す部分である。

ステップＳ４１では、ＵＲＬに記載された「＊」の位置が確定される。次いで、ステップＳ４２では、「＊」以降に記載されている、本来ユーザに見せたい方のＵＲＬ（この場合では「ｈｔｔｐ：／ｗｗｗ．３６５ｐｈａｒｍ１．ｃｏｍ／」の部分）を抽出する処理を行う。

ＵＲＬが抽出されたら、ＵＲＬの抽出・解析処理のステップＳ２６～ステップＳ２８において、ＵＲＬの解析が行われる。

【0055】

次に、図１５において、迷惑メールの判定処理及び迷惑メール送付遮断処理について説明する。判定処理は、図３に示すステップＳ５及びステップＳ６に該当する。

判定のフローは、ＭＴＡからローカルメイラーの代わりに、以下の処理を行うプログラムが呼び出されて行われる。

ステップＳ１及びステップＳ２の処理を経て、「メールの各パートのハッシュ値」、「復号後のＵＲＬ文字列」、「”ｈｏｓｔ：ｐｏｒｔ／ｐａｔｈ”文字列」、「ＩＰアドレス」が取得されると、DB_1～DB_4を参照し、一致するデータのスコアを計算し、スコアが１以上であるか否かが判定される（ステップＳ３１）。

スコアが１以上であった場合（ステップＳ３１；Ｙｅｓ）、迷惑メールであると判定され、迷惑メールボックス１３にメールが配送される（ステップＳ３２）。

また、スコアが０であった場合（ステップＳ３１；Ｎｏ）、迷惑メールではないと判定され、ユーザのメールボックス１２にメールが配送される（ステップＳ３３）。

このとき、メールのヘッダに検出結果（スコア）を付加して配送する。こうすることにより、ＭＵＡ（Ｍａｉｌ　Ｕｓｅｒ　Ａｇｅｎｔ）で、ヘッダ情報を利用した分別等の処理が可能となる。なお、配送先のアカウントを切り替える際は、図６の－ｕ及び－ｒオプションを使用することにより、切り替えが可能となる。

【0056】

図１６は、データベースDB_1～DB_4への登録処理を示すものである。この処理は、図３に示すステップＳ３に該当する。

登録のフローは、登録用のメールアドレスにメールを転送することで行われる。そうす

ると、登録用のプログラムが呼び出され、登録処理が行われる。

登録処理では、解析結果をもとに、「メールの各パートのハッシュ値」、「復号後のＵＲＬ文字列」、「”ｈｏｓｔ：ｐｏｒｔ／ｐａｔｈ”文字列」、「ＩＰアドレス」が、それぞれのデータベースDB_1～DB_4に登録される。

ステップＳ４１では、取得された上記４種類のデータについて、DB_1～DB_4が参照され、当該情報がDB_1～DB_4へ登録されているか否かが判定される。登録があった場合（ステップＳ４１；Ｙｅｓ）は、重複登録回数（カウンタ）を１増加させ、更新日時を更新して再登録する（ステップＳ４２）。

登録がなかった場合（ステップＳ４１；Ｎｏ）は新規登録となる。新規登録されるＵＲＬ情報は、重複登録回数（カウンタ）が１、更新日時は現在、アーカイブ名は現在処理中のもので登録される（ステップＳ４３）。

【0057】

図１７は、解析結果をユーザに返信する処理を示すものである。この処理は、図３におけるステップＳ４に該当する。

ステップＳ５１では、電子メールがテキストファイルとして出力される。出力されたデータは、「アーカイブ名．ｔｘｔ」というファイル名でデータ格納部１１ｂに格納される。

また、ステップＳ５２では、解析結果がＨＴＭＬ出力される。出力されたデータは、「アーカイブ名．ｈｔｍｌ」というファイル名でデータ格納部１１ａに格納される。

ステップＳ５３では、解析結果を閲覧希望するユーザに対して、これらの情報を閲覧するためのＵＲＬがメールにより送信される。テキストファイルは、解析結果のページからリンクを張って参照可能とされる。

【0058】

図１８及び図１９は、解析結果の一例を示すものである。

図１８に示す例では、送付された電子メールのタイプ（図中の符号Ａ）、本文のハッシュ値（図中の符号Ｂ）、各パートのメディアタイプ（図中の符号Ｃ）、各パートのハッシュ値（図中の符号Ｄ，Ｅ，Ｆ）、各パートに含まれていたＵＲＬの数（図中の符号Ｇ，Ｈ，Ｉ）、送付時のＵＲＬ文字列（図中の符号Ｊ）、復号後のＵＲＬ文字列（図中の符号Ｋ）、ＵＲＬを分解した結果情報（図中の符号Ｌ）、”ｈｏｓｔ：ｐｏｒｔ／ｐａｔｈ”の文字列（図中の符号Ｍ）、ＩＰアドレス（図中の符号Ｎ）等の情報が表示されている。

図１９に示す例では、送付された電子メールの解析結果に加えて、DB_1～DB_4に登録されるデータについても表示されている。

図１９では、送付された電子メールのタイプ（図中の符号Ａ）、本文のハッシュ値（図中の符号Ｂ）、メールに含まれていたＵＲＬ情報の数（図中の符号Ｇ）、送付時のＵＲＬ文字列（図中の符号Ｊ）、復号後のＵＲＬ文字列（図中の符号Ｋ）、ＵＲＬを分解した結果情報（図中の符号Ｌ）、”ｈｏｓｔ：ｐｏｒｔ／ｐａｔｈ”の文字列（図中の符号Ｍ）、ＩＰアドレス（図中の符号Ｎ）が表示されている。

さらに、抽出されたＩＰアドレスの重複登録回数（図中の符号Ｏ）、更新日時（図中の符号Ｐ）、アーカイブ名（図中の符号Ｑ）が表示されている。

【0059】

上記解析結果では、「ＮＥＷＵＲＬ」等、ＮＥＷの付いているデータが、新規登録されたことを意味しているので、ユーザはその内容をチェックする。

このとき、ＤＢに登録したくないＵＲＬ情報（すなわち、迷惑メールとは無関係なＵＲＬ情報）があれば、そのＵＲＬ情報はホワイトリストに登録される。

この場合は、迷惑メールと無関係なＵＲＬ情報をメールの本文に記載し、ホワイトリスト登録用のメールアドレスにメールを送信する。そうすると、図７の「ホワイトリスト登録とレポート」に記載されたプログラムが起動される。そして、該当情報がデータベースDB_1～DB_4から削除されるとともに、ホワイトリストのデータベースW_1～W_4への登録が行われる。

【0060】

ログファイルを作成する際は、ログファイルにはそのメールの含んでいるＵＲＬ情報とその解析結果及び、詳細なスコア（ＤＢにヒットした数／検出数）が記録される。

ログファイルは、例えば次のような形式で記録される。

Ｘ－Ｐｉｃｋｙ－Ｓｃｏｒｅ：１０１（ｉｐ：２３／２７、ｈｐｐ：３９／４３、ｕｒｌ：３９／４３、ｐｓｉｇ：０／３）

上記の例では、メールサーバで、同じＩＰアドレス（ｉｐ）、または同じｈｏｓｔｎａｍｅ：ｐｏｒｔ／ｐｔｈ文字列（ｈｐｐ）、または同じＵＲＬ（ｕｒｌ）、または同じ各パートの電子指紋（ｐｓｉｇ）のうち、全部で１０１個がDB_1～DB_4に登録されていたことを示している。

この例では、ｉｐ（ＩＰアドレス）については２７個検出のうちの２３個がDB_3の情報に一致し、ｈｐｐ（「ｈｏｓｔ：ｐｏｒｔ／ｐａｔｈ」文字列）についは４３個検出のうちの３９個がDB_2の情報に一致し、ｕｒｌ（復号後のＵＲＬ文字列）については４３個検出のうちの３９個がDB_1の情報に一致し、ｐｓｉｇ（メールの各パートのハッシュ値）については３個検出のうちの０個がDB_4の情報に一致したことが示されている。

【産業上の利用可能性】

【0061】

本例のシステムを、有害ウェブサイトへのアクセス制限に用いても良い。

この場合も、上記実施の形態と同様にして、有害ウェブサイトに関するＵＲＬ情報を取得する。このＵＲＬ情報を有する端末では、登録されたＵＲＬ情報に基づいて、有害ウェブサイトへのアクセスがなされようとしている場合に、そのアクセスが制限される。

【0062】

上記構成から把握できる請求項以外の技術的思想を以下に記載する。

（１）電子メールにより誘導される特定ウェブサイトへのアクセス制限方法であって、

前記電子メールを、メール本文に含まれるセパレータ文字列に従って、メール構造の最小単位であるパートに分解するメール解析工程と、

前記分解したパート毎に電子指紋を取得する電子指紋取得工程と、

前記パートに含まれるＵＲＬ情報を取得するＵＲＬ情報取得工程と、

過去の電子メールから取得・蓄積された電子指紋またはＵＲＬ情報が格納されたデータベースを参照するデータベース参照工程と、

前記電子指紋取得工程で取得された電子指紋または前記ＵＲＬ情報取得工程で取得されたＵＲＬ情報が、前記データベースに迷惑メールに関連する電子指紋またはＵＲＬ情報として格納されていた場合に、前記ＵＲＬ情報を有害ウェブサイトに関する情報と判定して、該有害ウェブサイトへのアクセスを制限するアクセス制限工程と、を備えたことを特徴とする特定ウェブサイトへのアクセス制限方法。

【実施例】

【0063】

迷惑メールに関するデータを提供するサイト（例えばスパムアーカイブ；http://www.spmarchive.org/）を利用して、本システムの機能を評価した。

迷惑メールを１通ずつ解析し、迷惑メールとして登録するという手順で、３４日分、３７，０００通あまりをＤＢに登録した。

その結果、ＵＲＬからＩＰアドレスが取得できたものに限ると、３１日目のデータで９９．５８％の迷惑メールを遮断することができた。これは一般的な迷惑メール遮断の手法を上回る検出率である。

【0064】

また、３７，０００通のメールを解析した結果、各ＤＢに記録されているデータエントリーは、

ＵＲＬ文字列のＤＢ：６８，４７３件

ｈｏｓｔ：ｐｏｒｔ／ｐａｔｈ文字列のＤＢ：４０，５２６件

メールの各パートのハッシュ値のＤＢ：３７，４２４件

ＩＰアドレスのＤＢ：５，２０２件

となった。

このなかで、検出に最も貢献したのは、ＩＰアドレスのＤＢであった。

このように、本例のシステムでは、小さなサイズのＤＢで高い検出効率を達成することができるため、大規模サイトへの適用も可能である。

また、従来の仕組みのように、発信元のホストやＩＰアドレス、発信者のドメインを利用して迷惑メールを判定する手法では、迷惑メールの発信元として登録されてしまうと、それ以降その発信元からはメールが届かなくなってしまうが、本例のシステムでは、誘導先のＵＲＬ情報さえ消せば確実にメールは到達するので、実用上、より弊害の少ない安全なシステムになっていると言える。

【図面の簡単な説明】

【0065】

【図1】一般的な電子メール送受信のシステム構成を示す説明図である。

【図2】本発明の実施の形態における迷惑メール遮断方法及びプログラムが実行されるシステムの構成図である。

【図3】迷惑メール検出の手順を示すブロック図である。

【図4】メール受信からメールボックス格納までの流れを示す説明図である。

【図5】メール受信から解析結果提示までの流れを示す説明図である。

【図6】プログラムのオプションの例を示す一覧である。

【図7】実際のａｌｉａｓの例を示す一覧である。

【図8】ＭＩＭＥメールの最小単位になるパートを示す説明図である。

【図9】マルチパートの構造からなるＭＩＭＥメールを示す説明図である。

【図10】マルチパートの構造からなるＭＩＭＥメールを示す説明図である。

【図11】マルチパートの電子メールの復号化処理を示すフローチャートである。

【図12】ＵＲＬの抽出・解析処理を示すフローチャートである。

【図13】符号化された文字の一覧表である。

【図14】符号化されたＵＲＬと復号化されたＵＲＬの例を示す説明図である。

【図15】迷惑メールの判定処理を示すフローチャートである。

【図16】データベースへの登録処理を示すフローチャートである。

【図17】解析結果をユーザに返信する処理を示すフローチャートである。

【図18】解析結果の一例を示す説明図である。

【図19】解析結果の一例を示す説明図である。

【図20】解析結果を保存する記憶部の構造を示す説明図である。

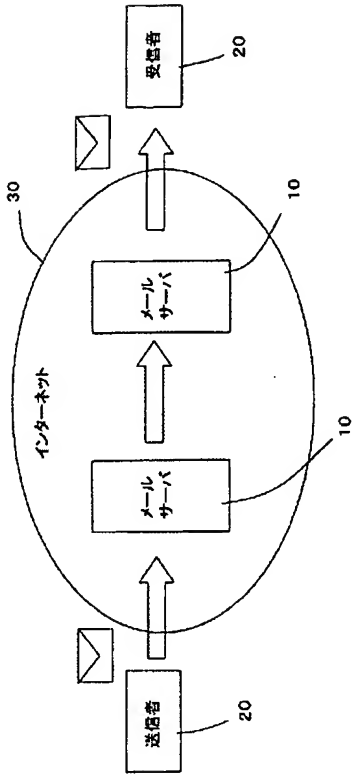
【図21】ＵＲＬに使用可能な文字の具体例を示す説明図である。

【符号の説明】

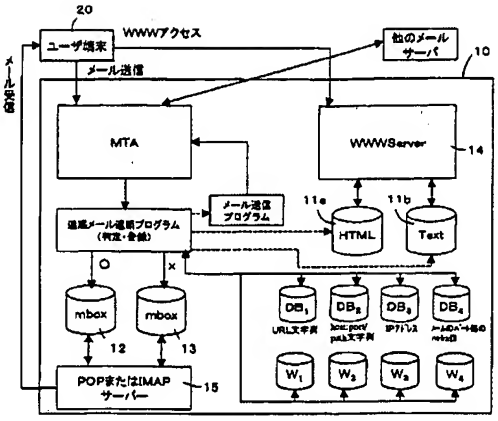
【0066】

１０　メールサーバ、１１　メール受信部、１２制御部、１３　記憶部、１４　メールボックス、１５　迷惑メールボックス、１６　リモート処理部、２０　ユーザ端末、３０　インターネット

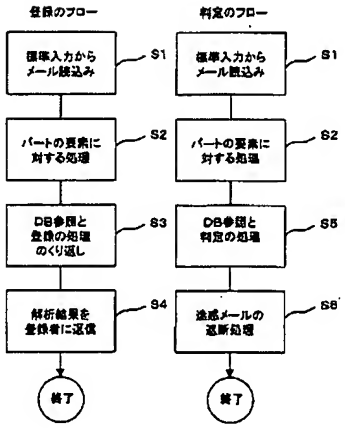
【図1】


受信者
20
30
10
メールサーバ
インターネット
メールサーバ
10
送信者
20

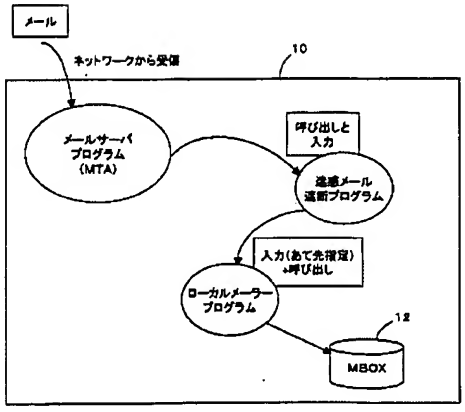
【図2】


20
ユーザ端末
WWWアクセス
他のメールサーバ
10
メール送信
MTA
WWWServer
14
メール送信プログラム
11a
11b
HTML
Text
O
x
mbox
mbox
12
13
POPまたはIMAPサーバー
15

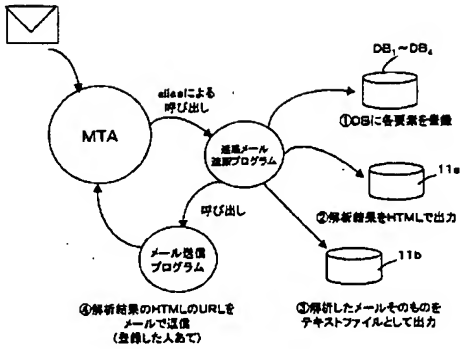
【図3】


登録のフロー
判定のフロー
標準入力からメール読込み
S1
パートの要素に対する処理
S2
DB参照と登録の処理のくり返し
S3
解析結果を登録者に返信
S4
終了
標準入力からメール読込み
S1
パートの要素に対する処理
S2
DB参照と判定の処理
S5
迷惑メールの遮断処理
S6
終了

【図4】


メール
ネットワークから受信
10
メールサーバプログラム（MTA）
呼び出しと入力
迷惑メール遮断プログラム
入力（あて先指定）+呼び出し
ローカルメーラープログラム
12
MBOX

【図5】



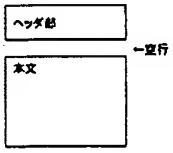
【図6】

```
-a: write results to html and report
-b: update DBs with analysis result
-c: check DBs (use this with -t or -u -r)
-d:<n>    :set debug level
-e: erase DB entries and write to html
-w: update white-list DBs
-t<user>:      all articles to user by mail, local
-u<user>:      passed article to user by mail, local
-r<user>:      rejected article to user by mail, local
-l<logfile>:   set logfile
-v: verbose command line output of analyze result
-q: quiet mode
```

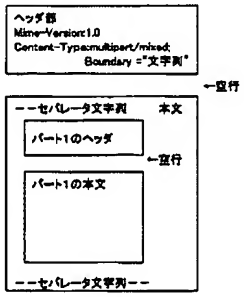
【図7】

```
#実配送モード
picky-test: "/usr/local/ppml/picky -e -q -t picky -l /var/log/picky. log"
#登録とレポート
picky-entry: "/usr/local/ppml/picky -d 9 -a -b -q -l /var/log/picky. log"
#解析とレポート
picky-check: "/usr/local/ppml/picky -d 9 -a -q -l /var/log/picky. log"
#削除とレポート
picky-erase: "/usr/local/ppml/picky -d 9 -a -e -q -l /var/log/picky. log"
#ホワイトリスト登録とレポート
picky-write: "/usr/local/ppml/picky -d 9 -a -e -w -q -l /var/log/picky. log"
```

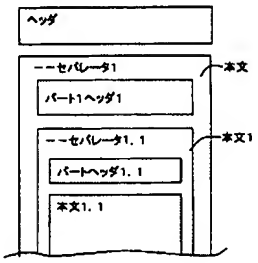
【図8】



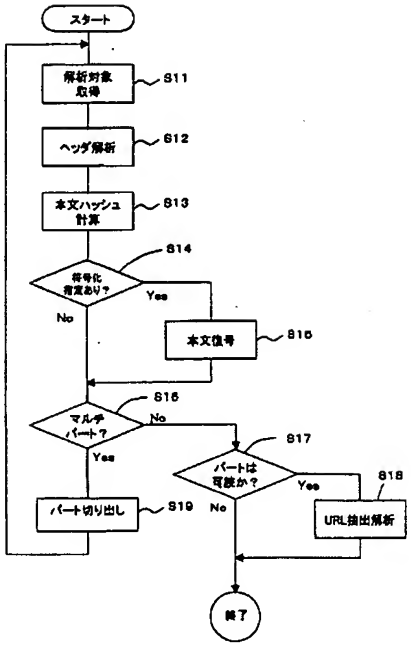
【図9】



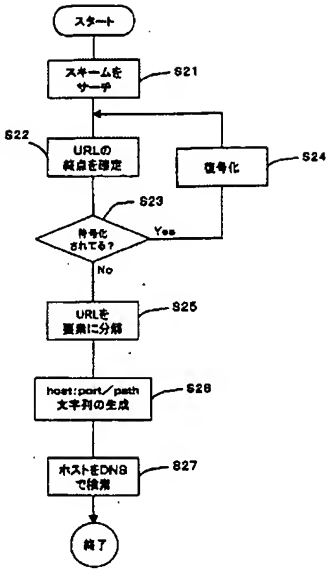
【図10】



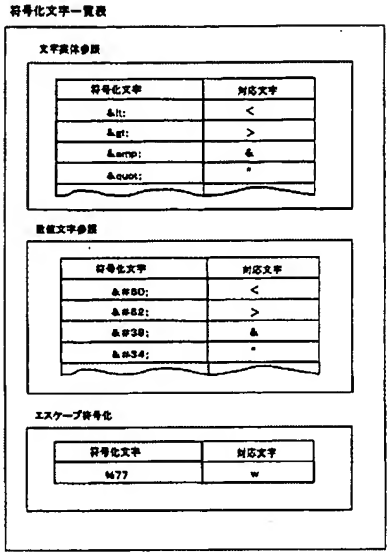
【図11】



【図12】



【図13】



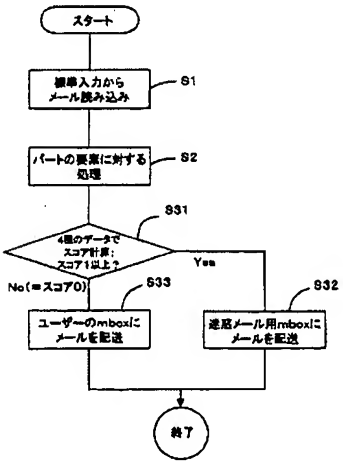
【図14】

(例1)
http://BuyhgHNoWCoM/a.html
http://BuyhgtNoW.Com/a.html

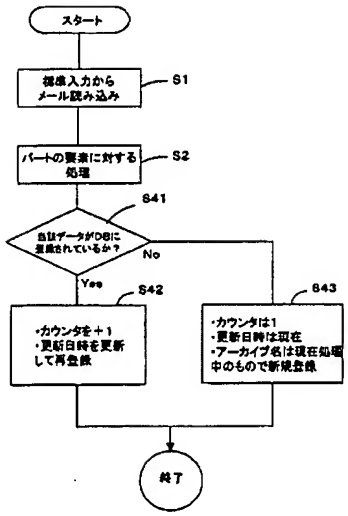
(例2)
http://&#66&#117&#121&#104&#103&#72&#78&#111&#87.&#67&#111&#77/b.html
http://BuyhgtNoW.Com/b.html

(例3)
http://treasure:wink@%6D%65%64s2%347%2E%69%6Efo/cd/ads.jpg
http://treasure:sink@meds247.info/cd/ads.jpg

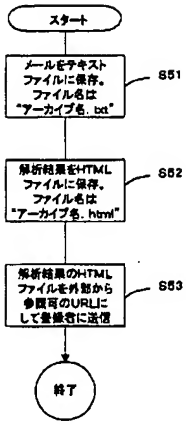
【図15】



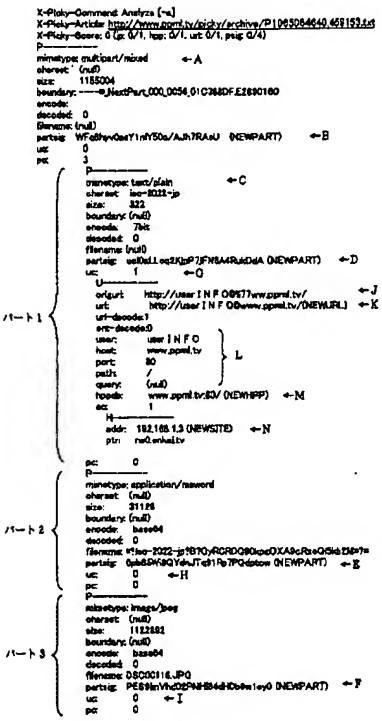
【図16】



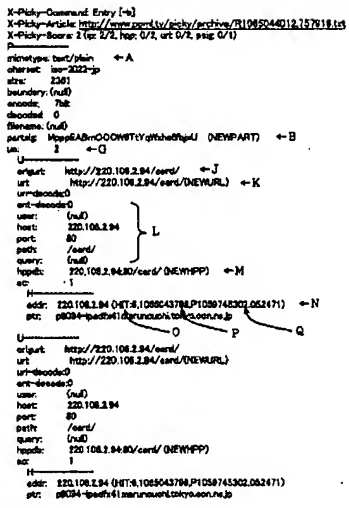
【図17】



【図18】



【図19】

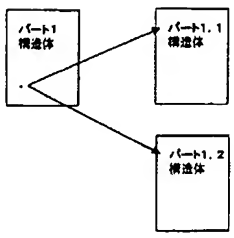


【図20】

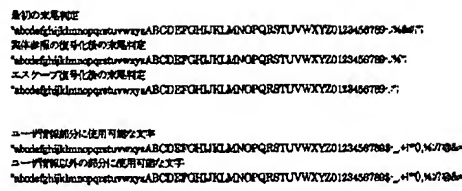
(a)

```
/* for multipart analysis */
struct mimepartsf
  char *mimetype;
  size_t size;
  char *charset;
  char *encode;
  int  decoded;
  char *boundary;
  char *body;
  char *filename;
  char *partsig;
  int  uri;
  struct linkuris uris[4096];
  int  pc;
  struct mimeparts *parts[256];
};
```

(b)



【図21】


最初の末尾判定
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789-.%&=?";
実体参照の復号化後の末尾判定
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789-.%";
エスケープ復号化後の末尾判定
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789-.";
ユーザ情報部分に使用可能な文字
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789$-_.+!*'(),%/?@&=~`[]\\n";
ユーザ情報以外の部分に使用可能な文字
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789$-_.+!*'(),%/?@&=~`;